Opt2Open
Promoting you in Open Source

Auditd : monitoring de données en temps réel

Auditd

Auditd est un outil qui permet de monitorer les accès aux données et de pouvoir offrir un support stable pour l'exploitation des logs. Auditd fait partie d'un ensemble de composants qui ont pour fonctionnalités de gérer les règles de surveillance et d'afficher les rapports.

Auditd est le daemon qui tourne en fond, en userspace, et qui est au coeur de la récupération des informations.

Prise en main

L'installation est très facile, et comme l'outil a fait ses preuves, on peut l'installer directement depuis les dépôts officiels.

```
[root@dlp ~]# yum -y install audit
[root@dlp ~]# service auditd start

[root@dlp ~]# systemctl enable auditd
```

Configuration

La configuration d'audit se trouve sous /etc/audit/. On y trouve auditd.conf, qui est la conf à proprement parler du daemon auditd et audit.rules qui est destiné à contenir vos règles en dur.

```
1 # ls -asl /etc/audit/
2 total 24
3 4 drwxr-x--- 2 root root 4096 2011-09-02 15:13 ./
4 12 drwxr-xr-x 140 root root 12288 2011-09-05 15:23 ../
5 4 -rw-r----- 1 root root 680 2009-07-27 21:07 auditd.conf
6 4 -rw-r----- 1 root root 373 2009-07-27 21:07 audit.rules
```

La conf ressemble à ça :

```
01 #
02 # This file controls the configuration of the audit daemon
03 #
04
05 log_file = /var/log/audit/audit.log
06 log_format = RAW
07 log_group = root
08 priority_boost = 4
09 flush = INCREMENTAL
10 freq = 20
11 num_logs = 4
12 disp_qos = lossy
13 dispatcher = /sbin/audispd
14 name_format = NONE
```

Opt2Open
Promoting you in Open Source

```
15 ##name = mydomain
16 max_log_file = 5
17 max_log_file_action = ROTATE
18 space_left = 75
19 space_left_action = SYSLOG
20 action_mail_acct = root
```

Cette conf nous apprend notamment l'emplacement du log généré par le daemon, ce qui nous permettra d'y jeter un œil plus tard, pour extraction.

Quant à *audit.rules*, à la base, il est un peu vide, comme vous pouvez le constater : un -D pour tout flusher, une instruction d'augmentation de buffer, en dehors de cela, tout reste à faire.

```
01 $ cat /etc/audit/audit.rules
02
03 # This file contains the auditctl rules that are loaded
04 # whenever the audit daemon is started via the initscripts.
05 # The rules are simply the parameters that would be passed
06 # to auditctl.
07
08 # First rule - delete all
09 -D
10
11 # Increase the buffers to survive stress events.
12 # Make this bigger for busy systems
13 -b 320
14
15 # Feel free to add below this line. See auditctl man page
```

Il y a donc 2 manières différentes de faire des règles : en statique dans le fichier de conf */etc/audit/audit.rules*, ou à la volée.

Bien sûr, les seules différences entre statique et à la volée sont que :

- la volée ne survit pas à un reboot, alors que statique se lancera avec le daemon auditd
- la syntaxe des règles statiques est la même que celles à la volée, mis à part l'invocation de **auditctl**.

Création et gestions des règles

Comme c'est plus facile, et que c'est pratique, nous allons créer quelques règles à la volée.

Pour information, voici les options que nous allons utiliser :

- w : watch, activer la surveillance
- p war : fixer le filtre concernant les permissions pour la surveillance d'un fichier. Ici, r pour read, w pour write, x pour execute, a pour append.
- k passwd-file : clé unique permettant d'identifier l'objet de la requête

```
1 k-root # auditctl -w /etc/passwd -p war -k password-file
2 k-root # ausearch -f /etc/passwd
```

J'ai crée un watch sur le fichier */etc/passwd*, qui relèvera toute tentative en read, write ou append sur le fichier.

Comme on peut le voir, pour le moment, aucun évènement n'a eu lieu… Mais c'est sans compter sur *k-user*, qui va *grepper* **root** sur ledit fichier…

```
1 k-user $ grep root /etc/passwd
2 root:x:0:0:root:/root:/bin/bash
```

Immédiatement, la tentative -fructueuse- est journalisée et vue par auditd :

```
01 k-root # ausearch -f /etc/passwd
02 ----
03 time->Fri Sep 2 15:16:50 2011
04 type=PATH msg=audit(1314969410.095:9): item=0 name="/etc/passwd"
inode=1982554 dev=08:01 mode=0100644 ouid=0 ogid=0 rdev=00:00
05 type=CWD msg=audit(1314969410.095:9): cwd="/home/k-user/Documents"
06 type=SYSCALL msg=audit(1314969410.095:9): arch=40000003 syscall=5
success=yes exit=4 a0=b763b078 a1=80000 a2=1b6 a3=8295090 items=1 ppid=6893
pid=28672 auid=4294967295 uid=3100 gid=3000 euid=3100 suid=3100 fsuid=3100
egid=3000 sgid=3000 fsgid=3000 tty=(none) ses=4294967295 comm="ps"
exe="/bin/ps" key="password-file"
07 ----
08 time->Fri Sep 2 15:16:52 2011
09 type=PATH msg=audit(1314969412.095:10): item=0 name="/etc/passwd"
inode=1982554 dev=08:01 mode=0100644 ouid=0 ogid=0 rdev=00:00
10 type=CWD msg=audit(1314969412.095:10): cwd="/home/k-user/Documents"
11 type=SYSCALL msg=audit(1314969412.095:10): arch=40000003 syscall=5
success=yes exit=4 a0=b7669078 a1=80000 a2=1b6 a3=86c6090 items=1 ppid=6893
pid=28673 auid=4294967295 uid=3100 gid=3000 euid=3100 suid=3100 fsuid=3100
egid=3000 sgid=3000 fsgid=3000 tty=(none) ses=4294967295 comm="ps"
exe="/bin/ps" key="password-file"
12 ---- [...]
```

On peut faire aussi un peu plus lisible, grâce à l'option -i qui interprètera les id (uid, guid, date, …). Plus facile pour le coup d'œil, n'est-ce pas ?

```
01 k-root # ausearch -f /etc/passwd -i
02 ----
03 type=PATH msg=audit(02/09/2011 15:16:50.095:9) : item=0 name=/etc/passwd
inode=1982554 dev=08:01 mode=file,644 ouid=root ogid=root rdev=00:00
04 type=CWD msg=audit(02/09/2011 15:16:50.095:9) : cwd=/home/k-
user/Documents
05 type=SYSCALL msg=audit(02/09/2011 15:16:50.095:9) : arch=i386
syscall=open success=yes exit=4 a0=b763b078 a1=80000 a2=1b6 a3=8295090
items=1 ppid=6893 pid=28672 auid=unset uid=k-user gid=k-user euid=k-user
suid=k-user fsuid=k-user egid=k-user sgid=k-user fsgid=k-user tty=(none)
ses=4294967295 comm=ps exe=/bin/ps key=password-file
06 ----
07 type=PATH msg=audit(02/09/2011 15:16:52.095:10) : item=0
name=/etc/passwd inode=1982554 dev=08:01 mode=file,644 ouid=root ogid=root
rdev=00:00
08 type=CWD msg=audit(02/09/2011 15:16:52.095:10) : cwd=/home/k-
user/Documents
09 type=SYSCALL msg=audit(02/09/2011 15:16:52.095:10) : arch=i386
syscall=open success=yes exit=4 a0=b7669078 a1=80000 a2=1b6 a3=86c6090
items=1 ppid=6893 pid=28673 auid=unset uid=k-user gid=k-user euid=k-user
```

Opt2Open
Promoting you in Open Source

```
suid=k-user fsuid=k-user egid=k-user sgid=k-user fsgid=k-user
tty=(none) ses=4294967295 comm=ps exe=/bin/ps key=password-file
10 ----
```

Les commandes de base sont intuitives, par exemple, le listing des règles en place :

```
1 k-root # auditctl -l
2 LIST_RULES: exit,always watch=/etc/passwd perm=rwa key=password-file
3 LIST_RULES: exit,always watch=/etc/shadow perm=rwax key=shadow-file
```

La suppression d'une règle (attention, une, et autant vous dire que les doublons ne sont pas conseillés).

En parlant de doublon, étant donné que l'évaluation d'une règle est parfois coûteuse, comme par exemple l'évaluation de tous les **syscall**, la factorisation d'expression est préconisée.

```
1 k-root # auditctl -l
2 LIST_RULES: exit,always watch=/etc/shadow perm=rwxa key=shadow-file
3 LIST_RULES: exit,always watch=/etc/passwd perm=rwa key=passwd-file
4
5 k-root # auditctl -d exit,never -W /etc/shadow -k shadow-file
6
7 k-root # auditctl -l
8 LIST_RULES: exit,always watch=/etc/passwd perm=rwa key=passwd-file
```

A titre d'information, le delete all a déjà été vu dans le fichier de règles : l'option en question est -D. Bon à savoir quand on veut resetter notre daemon.

Reporting

Pour ce qui est du reporting, pas de problème non plus, un tas d'options permettant en plus de mettre le focus sur ce que vous avez vraiment envie de consulter.

Pour savoir un peu ce qui tourne, on appellera ***auditctl***.

```
1 k-root # auditctl -s
2 AUDIT_STATUS: enabled=1 flag=1 pid=28521 rate_limit=0 backlog_limit=64
lost=0 backlog=0
```

Alors que pour le reporting à proprement parler, on passera par ***aureport*** : ici pour les évènements…

```
1 k-root # aureport -e
2 1840. 02/09/2011 16:14:25 1846 USER_AUTH -1 yes
3 1841. 02/09/2011 16:14:25 1847 CRED_REFR -1 yes
4 1842. 02/09/2011 16:14:23 1840 SYSCALL -1 yes
```

Et là pour lister les authentifications, réussi ou pas d'ailleurs.

```
1 k-root # aureport --auth
2
3 Authentication Report
4 ============================================
5 # date time acct host term exe success event
```

```
6 ============================================
7 1. 02/09/2011 15:16:46 root ? pts/26 /bin/su yes 4
8 2. 02/09/2011 16:14:25 k-user ? :0
/usr/lib/kde4/libexec/kcheckpass yes 1846
```

Et enfin, vue d'ensemble, à la logwatch :

```
01 # aureport --summary
02
03 Summary Report
04 ======================
05 Range of time in logs: 02/09/2011 15:14:10.327 - 02/09/2011 16:17:06.165
06 Selected time for report: 02/09/2011 15:14:10 - 02/09/2011 16:17:06.165
07 Number of changes in configuration: 4
08 Number of changes to accounts, groups, or roles: 0
09 Number of logins: 0
10 Number of failed logins: 0
11 Number of authentications: 2
12 Number of failed authentications: 0
13 Number of users: 1
14 Number of terminals: 8
15 Number of host names: 1
16 Number of executables: 23
17 Number of files: 2
18 Number of AVC's: 0
19 Number of MAC events: 0
20 Number of failed syscalls: 1
21 Number of anomaly events: 0
22 Number of responses to anomaly events: 0
23 Number of crypto events: 0
24 Number of keys: 2
25 Number of process IDs: 1847
26 Number of events: 1930
```

A titre d'indication, voilà ce que renvoie le help de *aureport*.

```
01 k-root # aureport --help
02 usage: aureport [options]
03 -a,--avc Avc report
04 --auth Authentication report
05 -c,--config Config change report
06 -cr,--crypto Crypto report
…
```

Conclusion

Auditd est un outil bien pratique, à portée de main, qui peut être appliqué dans un contexte de sécurisation et de monitoring comme de reporting sur des modifications effectuées sur les données du système.

Il permet d'avoir un état des lieux concis et indiscutable et de pouvoir remonter le fil des évènements sans devoir dépendre totalement des informations fournies (ou pas) par les auteurs des modifs.